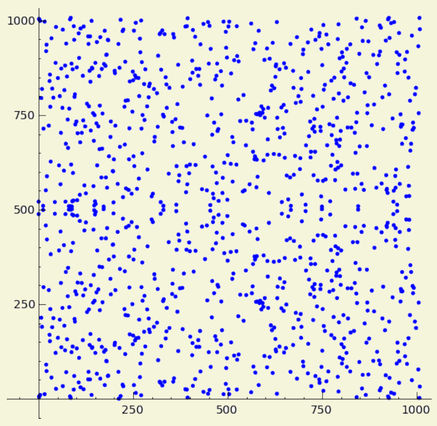
RNS Modular Computations for Cryptographic Applications

Karim Bigou & Arnaud Tisserand

1. Elliptic Curve Cryptography (ECC)

Elliptic curve over $\mathbb{F}_P$: $y^2 = x^3 + ax + b$ with P a ℓ-bit prime



$y^2 = x^3 + 4x + 20$ over $\mathbb{F}_{1009}$

Security levels: $\ell \in \{160, \ldots, 600\}$ bits

Curve level operations:

- point addition (ADD): $\mathbf{Q} + \mathbf{Q}'$
- point doubling (DBL): $\mathbf{Q} + \mathbf{Q}$
- scalar multiplication: $[k]\mathbf{Q} = \underbrace{\mathbf{Q} + \mathbf{Q} + \ldots + \mathbf{Q}}_{k \text{ times}}$

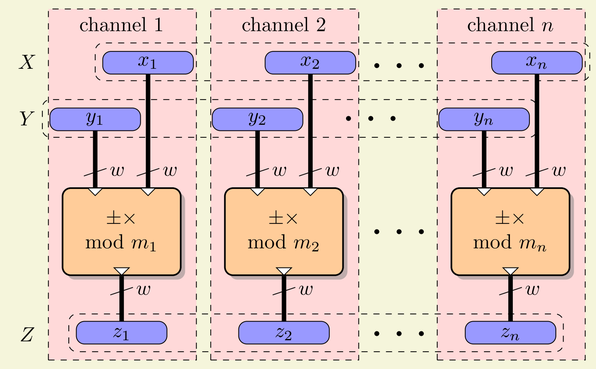
Security (ECDLP): knowing $\mathbf{Q}$ and $[k]\mathbf{Q}$, k cannot be recovered

ECDLP : Elliptic Curve Discrete Logarithm Problem

2. Residue Number System (RNS)

X a large ℓ-bit integer is represented by:

$$\vec{X} = (x_1, \ldots, x_n) = (X \bmod m_1, \ldots, X \bmod m_n)$$



RNS base $\mathcal{B} = (m_1, \ldots, m_n)$

n pairwise w-bit co-primes with $n \times w \geqslant \ell$

The Chinese remainder theorem (CRT) is the base of RNS

EMM elementary modular multiplication (w bits)

Pros:

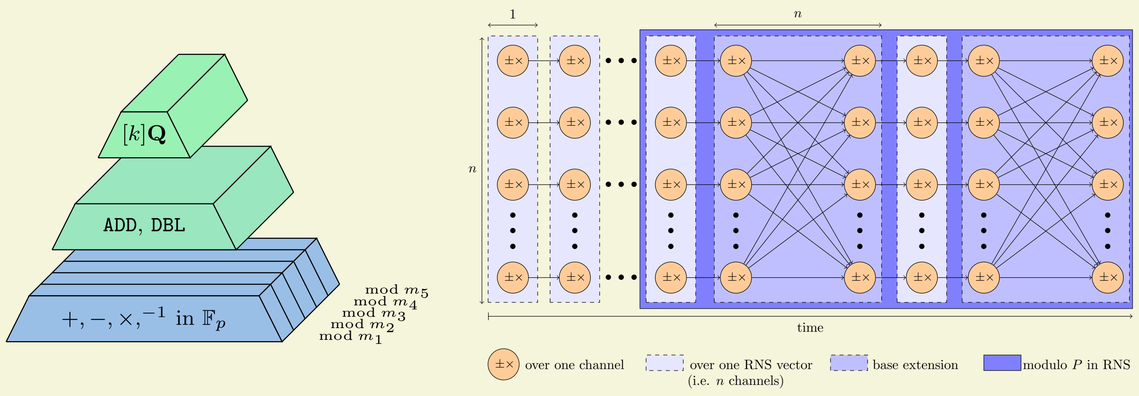
- **carry free** between channels
- **fast parallel** $+, -, \times$ and some exact divisions
- **non-positional** number system, randomization against SCAs
- **flexibility** for hardware implementations

Cons:

- comparison, **modular reduction** and division are much harder

3. RNS Computation Flow in ECC Applications

RNS allows to perform some field level operations in parallel



4. State-of-the-Art Algorithms and Architectures

RNS Montgomery Reduction

Input: $\vec{X}$, $\vec{X}'$

Output: $(\vec{\omega}, \vec{\omega}')$ with $\omega \equiv X \times M^{-1} \bmod P$

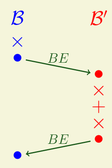
$\vec{Q} \longleftarrow \vec{X} \times (-\vec{P}^{-1})$ (in base $\mathcal{B}$)

$\vec{Q}' \longleftarrow \mathrm{BE}(\vec{Q}, \mathcal{B}, \mathcal{B}')$

$\vec{S}' \longleftarrow \vec{X}' + \vec{Q}' \times \vec{P}'$ (in base $\mathcal{B}'$)

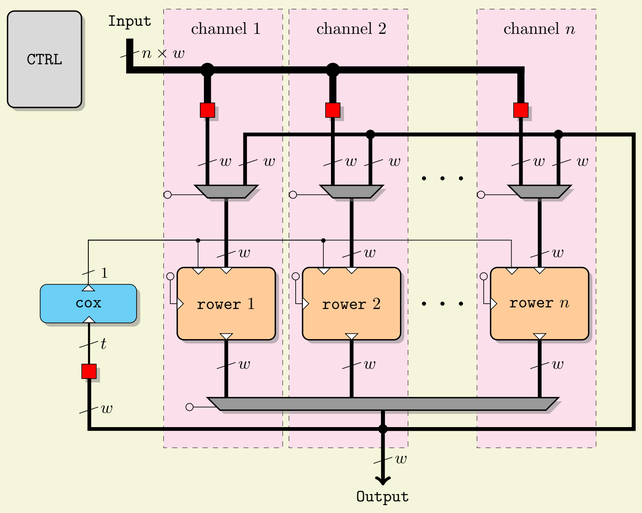
$\vec{\omega}' \longleftarrow \vec{S}' \times \vec{M}^{-1}$ (in base $\mathcal{B}'$)

$\vec{\omega} \longleftarrow \mathrm{BE}(\vec{\omega}', \mathcal{B}', \mathcal{B})$



BE: base extension

$M = \prod m_i$



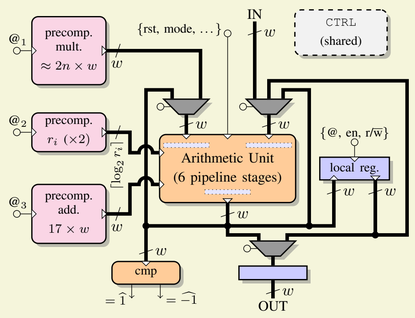
5. New RNS Modular Inversion (MI) (CHES 2013)

State-of-the-art RNS MI methods:

- based on Fermat's Little Theorem (FLT-MI): $X^{-1} = X^{P-2} \bmod P$ *i.e.* a large exponentiation with a lot of modular reductions which costs $O(\log_2 P \times n^2)$ EMMs
- very limited parallelization due to internal data dependencies

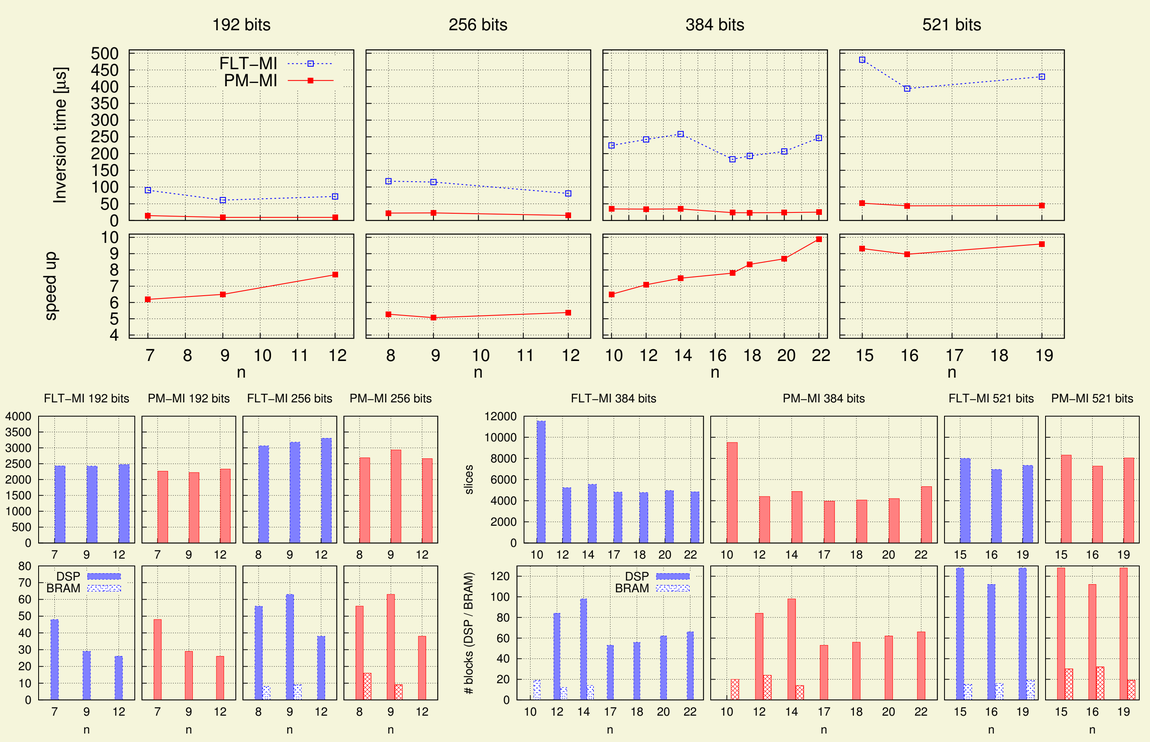
Proposed method PM-MI:

- extended binary Euclidean algorithm (binary-ternary version)
- uses the plus-minus trick: if X and Y are odd then $X + Y = 0 \bmod 4$ or $X - Y = 0 \bmod 4$
- PM-MI works without BE and costs $O(\log_2 P \times n)$ EMMs



Example: # EMMs for $\ell = 192$ bits

$n \times w$	FLT-MI	PM-MI	Gain Factor
12×17	103140	5474	18
9×22	61884	4106	15
7×29	40110	3193	12



6. Fast Patterns for RNS Computations (ASAP 2014)

Cost of standard and modular multiplications in RNS:

- standard: n EMMs — fully parallel
- modular: $2n^2 + O(n)$ EMMs — 1 mult. & 1 red.

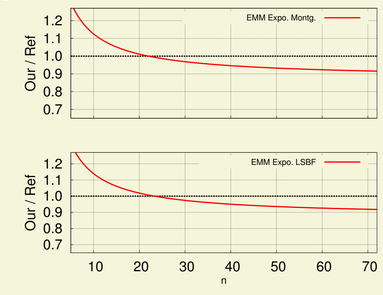
Proposed method:

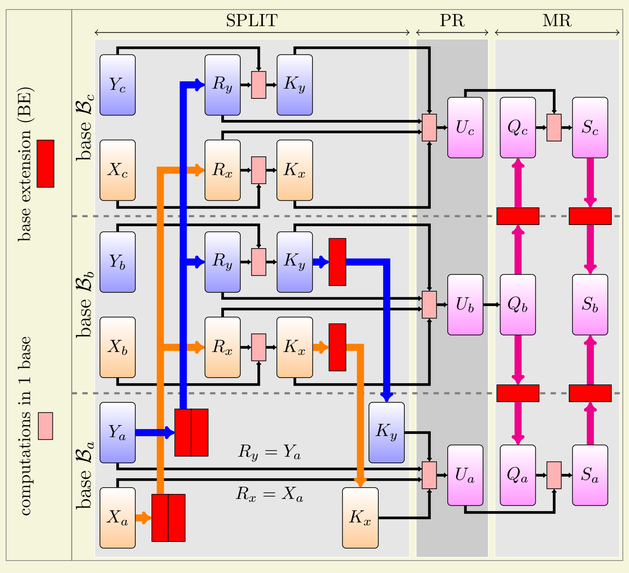
- splits operands into 2 parts: $\vec{X} = \overrightarrow{(K_x)} \times \overrightarrow{(M_a)} + \overrightarrow{(R_x)}$ allows to replace $2n$ moduli by only $\frac{3}{2}n$
- reuses split result in various computation patterns
- requires an hypothesis on P: OK for ECC/DH, but not for RSA

Cost for some patterns (#EMMs):

Operations	s-o-t-a	our
$AB \bmod P$	$2n^2 + 4n$	$2.5n^2 + 12.5n$
$A^2 \bmod P$	$2n^2 + 4n$	$1.75n^2 + 10.5n$
$Cst \times A \bmod P$	$2n^2 + 4n$	$1.75n^2 + 7n$
$Cst \times A^2 \bmod P$	$4n^2 + 8n$	$2.75n^2 + 16.5n$

Usage for Diffie-Hellman or ElGamal:





Funding from DGA-INRIA PhD grant and project PAVOIS ANR 12 BS02 002 01

http://pavois.irisa.fr/